

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method of autonomically patching computer program code, comprising:
checking a flag in a machine status register to determine whether code patching functionality is to be enabled;
responsive to determining that code patching functionality is to be enabled, enabling the code patching functionality;
executing a computer program instruction, wherein the computer program instruction is located at a start of a block of code of an execution sequence of original code instructions;
determining whether metadata is associated with the computer program instruction, wherein the metadata identifies the computer program instruction as a computer program instruction having associated patch instructions, and indicates an address of the patch instructions;
responsive to determining that metadata is associated with the computer program instruction, redirecting execution to the patch instructions at the address indicated by the metadata;
executing the patch instructions;
returning to an instruction of the execution sequence of original code instructions in the computer program; and
storing a result of executing the execution sequence.
2. (Original) The method of claim 1, wherein the patch instructions are created during execution of the computer program.
3. (Previously presented) The method of claim 1, wherein the patch instructions are created by:
copying instructions from the block of code to a new memory location;
modifying the order of the instructions of the block of code; and
populating the metadata with a pointer to the patch instructions.
4. (Original) The method of claim 1, wherein the metadata is in a form of a memory word.
5. (Previously presented) The method of claim 1, wherein the metadata includes a pointer to the patch instructions for indicating the address of the patch instructions.

6. (Original) The method of claim 5, wherein the pointer to the patch instructions includes a starting address of the patch instructions in an allocated memory location.
7. (Original) The method of claim 6, wherein the starting address includes at least one of an absolute or offset address.
8. (Canceled)
9. (Previously presented) The method of claim 1, wherein the patch instructions includes at least one of reorganized instructions, instrumented alternative instructions, and hooks to build an instruction trace.
10. (Currently amended) A data processing system having a processor for autonomically patching computer program code, the data processing system comprising:
 - checking means for checking a flag in a machine status register to determine whether code patching functionality is to be enabled;
 - enabling means, responsive to determining that code patching functionality is to be enabled, for enabling the code patching functionality;
 - executing means for executing a computer program instruction, wherein the computer program instruction is located at a start of a block of code of an execution sequence of original code instructions;
 - determining means for determining whether metadata is associated with the computer program instruction;
 - redirecting means, responsive to determining that metadata is associated with the computer program instruction, for redirecting execution to the patch instructions at the address indicated by the metadata;
 - executing means for executing the patch instructions;
 - returning means for returning to an instruction of the execution sequence of original code instructions in the computer program; and
 - storing means for storing a result of executing the execution sequence.
11. (Previously presented) The data processing system of claim 10, wherein the patch instructions are created during execution of the computer program.

12. (Previously presented) The data processing system of claim 10, wherein the patch instructions are created by:

copying instructions from the block of code to a new-memory location;
modifying the order of the instructions of the block of code; and
populating the metadata with a pointer to the patch instructions.

13. (Previously presented) The data processing system of claim 10, wherein the metadata is in a form of a memory word.

14. (Previously presented) The data processing system of claim 10, wherein the metadata includes a pointer to the patch instructions for indicating the address of the patch instructions.

15. (Previously presented) The data processing system of claim 14, wherein the pointer to the patch instructions includes a starting address of the patch instructions in an allocated memory location.

16. (Previously presented) The data processing system of claim 15, wherein the starting address includes at least one of an absolute or offset address.

17. (Canceled)

18. (Previously presented) The data processing system of claim 10, wherein the patch instructions includes at least one of reorganized instructions, instrumented alternative instructions, and hooks to build an instruction trace.

19. (Currently amended) A computer program ~~product~~ product stored in a computer ~~readable~~ storage medium for when executed by a computer, autonomically patching computer program code, the computer program product comprising:

first instructions for checking a flag in a machine status register to determine whether code patching functionality is to be enabled;

second instructions, responsive to determining that code patching functionality is to be enabled, for enabling the code patching functionality;

[[first]] third instructions for executing a computer program instruction, wherein the computer program instruction is located at a start of a block of code of an execution sequence of original code instructions;

~~second~~ fourth instructions for determining whether metadata is associated with the computer program instruction;

responsive to determining that metadata is associated with the computer program instruction, ~~[[third]]~~ fifth instructions for redirecting execution to the patch instructions at the address indicated by the metadata;

~~fourth~~ sixth instructions for executing the patch instructions;

~~[[fifth]]~~ seventh instructions for returning to an instruction of the execution sequence of original code instructions in the computer program; and

~~[[sixth]]~~ eighth instructions for storing a result of executing the execution sequence.

20. (Previously presented) The computer program product of claim 19, wherein the patch instructions are created during execution of the computer program.

21. (Previously presented) The computer program product of claim 19, wherein the patch instructions are created by copying instructions from the block of code to a new memory location;

modifying the order of the instructions of the block of code; and

populating the metadata with a pointer to the patch instructions.

22. (Previously presented) The computer program product of claim 19, wherein the metadata is in a form of a memory word.

23. (Previously presented) The computer program product of claim 19, wherein the metadata includes a pointer to the patch instructions for indicating the address of the patch instructions.

24. (Previously presented) The computer program product of claim 23, wherein the pointer to the patch instructions includes a starting address of the patch instructions in an allocated memory location.

25. (Previously presented) The computer program product of claim 24, wherein the starting address includes at least one of an absolute or offset address.

26. (Canceled)

27. (Previously presented) The computer program product of claim 19, wherein the patch instructions includes at least one of reorganized instructions, instrumented alternative instructions, and hooks to build an instruction trace.